

IN THE CLAIMS:

Please amend the claims as follows:

1. (Currently Amended) A processor comprising:

a decoder to decode a load instruction, said load instruction to be memory renamed to use an address of a previous ~~source~~ store instruction;

a memory ordering buffer to maintain the address for the ~~source~~ previous store instruction, wherein the address for the ~~source~~ previous store instruction is to be de-allocated from the memory ordering buffer after completion of the ~~source~~ previous store instruction;

a trailing store buffer to maintain an address for said ~~source~~ previous store instruction in response to said ~~source~~ previous store instruction ~~has been~~ being de-allocated from said memory ordering buffer and [[if]] only if said ~~source~~ previous store instruction ~~was being~~ was being a source of memory renaming, said trailing store buffer to maintain the address for said ~~source~~ previous store instruction to disambiguate said load instruction.

2. (Currently Amended) The processor of claim 1 wherein said memory ordering buffer further comprises:

a store address buffer to maintain the address for said ~~source~~ previous store instruction.

3. (Currently Amended) The processor of claim 1 wherein said memory ordering buffer further comprises:

a store data buffer to maintain data associated with said ~~source~~ previous store instruction.

4. (Original) The processor of claim 1 further comprising:

a store data buffer coupled to said memory ordering buffer.

5. (Original) The processor of claim 1 wherein said trailing store buffer is coupled to said memory ordering buffer.

6. (Original) The processor of claim 1 wherein said memory ordering buffer comprises said trailing store buffer.

7. (Cancelled)

8. (Currently Amended) A method comprising:

computing a store address for a store instruction;

writing the store address in a first storage;

writing data associated with the store address to a memory;

de-allocating the store address from the first storage after completion of the store instruction;

allocating the store address in a second storage after de-allocating the store address from the first storage [[if]]only if the store instruction being was a source of memory renaming;

predicting a load instruction to be memory renamed;

computing a load store source index;

computing a load address;

disambiguating the memory renamed load instruction by determining whether the store address is still stored in the first storage and, if not, determining whether the store address is stored in the second storage; and

retiring the memory renamed load instruction, if the store address is still allocated in at least one of said first storage and said second storage.

9. (Original) The method of claim 8 wherein computing a store address comprises:  
computing an address for a store instruction.

10. (Original) The method of claim 8 wherein writing the store address in a first storage comprises:

writing the store address in a store address buffer.

11. (Original) The method of claim 10 wherein writing data associated with the store address to a memory comprises:

writing the data from said store data buffer to said memory using the store address in said store address buffer.

12. (Original) The method of claim 11 wherein said store data buffer is in the first storage.

13. (Original) The method of claim 11 wherein said store data buffer is external to the first storage.

14. (Original) The method of claim 8 wherein de-allocating the store address from the first storage comprises:

de-allocating the store address from a store address buffer in the first storage.

15. (Canceled)

16. (Currently Amended) The method of claim 8 further comprises:  
determining whether said ~~source~~-store address for the memory renamed load instruction is in the second storage.

17. (Currently Amended) The method of claim 8 further comprising:  
clearing a backend of the processor and restarting the load instruction without memory renaming, if said ~~source~~ store address has been de-allocated from said first storage and said second storage.

18. (Currently Amended) A machine-readable medium having stored thereon a plurality of executable instructions to perform a method comprising:  
computing a store address for a store instruction;  
writing the store address in a first storage;  
writing data associated with the store address to a memory;  
de-allocating the store address from the first storage after completion of the store instruction;

allocating the store address in a second storage after de-allocating the store address from the first storage [[if]]only if the store was being a source of memory renaming;

predicting a load instruction to be memory renamed;

computing a load store source index;

computing a load address;

disambiguating the memory renamed load instruction by determining whether the store address is still stored in the first storage and, if not, determining whether the store address is stored in the second storage; and

retiring the memory renamed load instruction, if the store address is still allocated in at least one of said first storage and said second storage.

19. (Original) The machine-readable medium of claim 18 wherein computing a store address comprises:

computing an address for a store instruction.

20. (Original) The machine-readable medium of claim 18 wherein writing the store address in a first storage comprises:

writing the store address in a store address buffer.

21. (Original) The machine-readable medium of claim 20 wherein writing data associated with the store address to a memory comprises:

writing the data from said store data buffer to said memory using the store address in said store address buffer.

22. (Original) The machine-readable medium of claim 21 wherein said store data buffer is in the first storage.

23. (Previously Presented) The machine-readable medium of claim 21 wherein said store data buffer is external to the first storage.

24. (Original) The machine-readable medium of claim 18 wherein deallocating the store address from the first storage comprises:

de-allocating the store address from a store address buffer in the first storage.

25. (Canceled)

26. (Currently Amended) The machine-readable medium of claim 18 further comprises: determining whether said ~~source~~ store address for the memory renamed load instruction is in the second storage.

27. (Currently Amended) The machine-readable medium of claim 18 further comprising: clearing a backend of the processor and restarting the load instruction without memory renaming, if said ~~source~~ store address has been de-allocated from said first storage and said second storage.

28. (Currently Amended) A computer system comprising:

a processor comprising:

a decoder to decode a load instruction naming a destination register;

a memory ordering buffer to maintain an address for a ~~source~~ store instruction, wherein the address for the ~~source~~ store instruction is to be de-allocated from the memory ordering buffer after completion of the ~~source~~ store instruction; and

a trailing store buffer to maintain an address for said ~~source~~ store instruction if in response to said ~~source~~ store instruction has been being de-allocated from said memory ordering buffer and [[if]]only if said store was being a source of memory renaming, said trailing store buffer to maintain the address for said ~~source~~ store instruction to disambiguate said load instruction; and

a memory coupled to said processor.

29. (Currently Amended) The computer system of claim 28 wherein said memory ordering buffer further comprises:

a store address buffer to maintain an address for said ~~source~~ store instruction.

30. (Currently Amended) The computer system of claim 28 wherein said memory ordering buffer further comprises:

a store data buffer to maintain data associated with said ~~source~~ store instruction.